面向中文敏感词变形体的识别方法研究 *

付　聪 [1]，余敦辉 [1,2†]，张灵莉 [1]

(1. 湖北大学 计算机与信息工程学院, 武汉 430062; 2. 湖北省教育信息化工程技术中心, 武汉 430062)

摘　要：为净化网络环境，需要对网络信息进行审查。针对网络信息中所包含的敏感词，尤其是中文敏感词变形体的识别成为了一个迫切需要解决的问题。通过分析汉字的结构和读音等特征提出了一种中文敏感词变形体的识别方法。该方法针对词的拼音、词的简称和词的拆分三种敏感词变形体分别设计了基于易混拼音分组的敏感词的识别算法（SPGR）、字符串的简称识别算法（SNR）和基于 KMP 的汉字拆分识别算法（WS-KMP），有效提高了敏感词审查的准确率和效率。实验结果表明，该方法在识别中文敏感词变形体的时候有较高的查全率和查准率。

关键词：变形体；敏感词识别；编辑距离；KMP 算法

中图分类号：TP391.1　　doi: 10.3969/j.issn.1001-3695.2017.11.0996

Study on identification method for change form of Chinese sensitive words

Fu Cong[1], Yu Dunhui[1, 2†], Zhang Lingli[1]

(1. *School of Computer Science & Information Engineering Hubei University*, *Wuhan* 430062, *China*; 2. *Hubei Provincial Center for Education Information Technology Studies*, *Wuhan* 430062, *China*)

Abstract: To purify the network environment, the network information needs to be reviewed. Recognizing the sensitive words in the network information, especially the change form of Chinese sensitive words, is an urgent problem to be solved. By analyzing the structure and pronunciation of Chinese characters, this paper proposes a method of recognition of the change form of Chinese sensitive words. This method has designed sensitive word recognition algorithm based on the grouping of confusing pinyin, String abbreviation recognition algorithm and recognition algorithm based on KMP's character split recognition algorithm for the pinyin of word , the abbreviation of word and the split of word, and improve the accuracy and efficiency of the review. The experimental results show that the proposed method has higher recall and precision when recognizing the change form of Chinese sensitive words.

Key Words: change form; sensitive word recognition; edit distance; KMP algorithm

0　引言

随着互联网的高速发展，各种各样的信息资源呈指数级增长，非法言论（如黄赌毒、恐怖、暴力血腥信息）经常充斥其中[1-2]，这些不良信息通常带有一些敏感词汇，从而很可能引起不良的连锁反映，对国家安全、社会稳定和网络环境的健康形成严重威胁，造成巨大的负面影响。因此，对于敏感词的识别已经成为一个迫切需要解决的研究课题。

当前，对敏感词的识别研究较为成熟，一般是基于敏感词表进行。基本思想是对待检测的文本进行检索，若其中含有敏感词，则系统会判定该文本为需要审查的文本。这种方法实现起来比较简单，但查找的效率不高。对此，文献[3]提出一种 ST-DFA 算法,通过敏感词拼音的第一个字母来构建敏感信息决策树，其优点是不依赖敏感信息语料库，能够提高敏感信息的检测效率。缺点是对敏感词变形体无处理能力。

当前，对中文敏感词变形体的识别方法的研究还处于起步阶段，相关研究成果并不是太多，文献[4]针对变异的敏感词汇提出了一种方法。将某特殊字符转换成形状相似的字母，然后再进行检测。例如：将字符“@”转换成字母“a”,遇到“@rse”词后，将这个词换成“arse”来处理。文献[5]采用机器学习的方法,通过采用 bigram、词干等作为特征值来对文本信息做分类分析，以检测出变形体。文献[6]利用贝叶斯滤波技术对恶意内容进行检测,利用近似的字符串匹配技术来提高检测恶意内容的有效性。文献[7]提出了一种基于语音的字符串匹配算法，该算法用于解决发音相似的字符串的匹配。这些方法对英文字符有较好的处理效果，但没有将中文敏感词变形体考虑在内。

由此可见，寻找更有效的中文敏感词变形体识别算法是当务之急。本文结合英文字符串变形体的识别方法，提出了一种

基金项目：国家“973”计划资助项目（2014CB340404）；国家自然科学基金资助项目（61373037，61672387）

作者简介：付聪（1991-），男，湖北武人，硕士研究生，主要研究方向为大数据；余敦辉（1974-），男（通信作者），副教授，博士，主要研究方向为服务计算、大数据（yumhy@163.com）；张灵莉（1993-），女，硕士研究生，主要研究方向为大数据.

中文敏感词变形体识别方法。该方法结合了汉字的发音与结构特征，在识别的过程中加入拼音、简称、拆分三种变形体形式，能够有效的识别出中文敏感词及其相关的变形体。

1　相关算法

现有的字符串识别算法有基于相似度的匹配方法和精准匹配的模式匹配算法。基于编辑距离的相似度的匹配方法能够很好体现出一个字符串变为另一个字符串所需的“代价”，“代价”越小相似度越高。精确匹配常用的是 KMP 算法，相较与朴素的模式匹配算法减少了字符串匹配次数。

1.1　基于编辑距离的相似度计算

1965 年俄罗斯科学家 Vladimir Levenshtein 提出了编辑距离（edit distance）概念[8],又称为 Levenshtein 距离，是指两个字符串之间，由一个转成另一个所需的最少编辑操作次数。许可的编辑操作包括将一个字符替换成另一个字符，插入一个字符，删除一个字符。编辑距离被经常用于计算两个字符串之间的差异程度，编辑距离越小两个字符串的相似度越大[9-10]。本文利用编辑距离计算相似度、并利用归一化方法将其映射到 $[0,1]$ 区间：

$$\mathrm{Similar}(s,a)=1-\frac{\mathrm{edit}(\mathrm{length}(s),\mathrm{length}(a))}{\max(\mathrm{length}(s),\mathrm{length}(a))}$$

其中：$\mathrm{edit}(\mathrm{length}(s),\mathrm{length}(a))$ 表示字符串 s 与 t 的编辑距离，$length(s)$ 表示字符串 s 的长度，$\max(\mathrm{length}(s),\mathrm{length}(a))$ 表示 $\mathrm{length}(s)$ 与 $\mathrm{length}(a)$ 中长度较大的部分。

1.2　KMP 算法

1969 年 Knuth、Morris 和 Pratt 提出快速单模式匹配 KMP 算法[11]。它的主要思想是：每当一次匹配过程中出现字符不匹配时，不需要回退指针，而是利用已经得到的“部分匹配”的结果将模式向右移动尽可能远的一段距离，继续匹配过程。

设目标串表示为 $S=S[s_1,s_2,\cdots,s_n]$，长度为 n；模式串表示为 $A=A[a_1,a_2,\cdots,a_n]$，长度为 m；并且满足条件 $n>m$。如果存在 i 使得 $S[i]=A[1]$，$S[i+1]=A[2]$，…，$S[i+m-1]=A[m]$，则匹配成功，模式串 P 就出现在目标串 T 的 i 处。若 $S[i]=A[j]$，则继续比较 $S[i+1]=A[j+1]$；若 $S[i]\neq A[j]$，则 i 值不变，j 为 $next(j)$，再进行下一轮的匹配。其中 $next(j)$ 的值表示 $[a_1,a_2,\cdots,a_{j-1}]$ 中后缀等于相同字符序列的前缀的最长后缀的长度，对 next 数组的定义如下：

$$Next(j)=\begin{cases}-1，当J=1时\\ max\{k\mid 0<k<j,且存在\}\\ 0，其他情况\end{cases}$$

在下一次匹配时只需确定 j 的位置即可，从而提高模式匹配的效率。

2　敏感词变形体

本文研究的敏感词变形体包括以下三个模式：一是词的拼音模式。在中文中，同一个汉语拼音可能对应多个不同的汉字。且由于不同省市地区方言的差异性，导致了一些带有地域性特点的汉语拼音容易被混淆，如某些地区很难分辨“l”和“n”。二个是词的简称模式。在人们的日常生活中，对于字数较多的词，经常会以简称的形式替代。据统计，在汉语新闻文章里，在 20%左右的句子可能含有缩略语[12]。第三个是词的拆分模式。由于现在各个网络平台对信息的审查越来越严格，网络上出现了通过拆字的方式来逃避审查，比如“林”可以拆成“木木”[13]。以词“贩卖毒品”为例，其变形体的具体结构如图 1 所示。

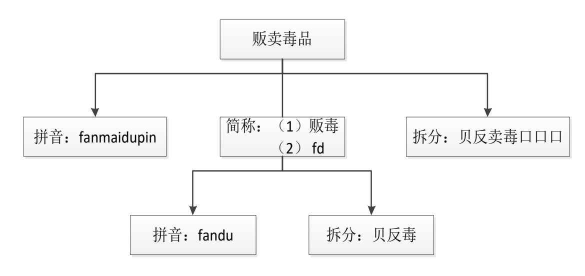


图 1　词变形体结构图

为了更好地识别出敏感词的变形体，本文基于已有的敏感词列表提出了敏感词变形体的识别算法（以下简称为 DFR（deformation form recognition）算法），对敏感词变形体进行识别。并且对不同的变形体形式给出了不同的处理方法，设计了一种基于易混拼音分组的敏感词识别算法（SPGR），通过相似度的计算识别出敏感词的拼音变形体，构建了字符串的简称识别算法（SNR），通过敏感词的首字母和缩写规则识别出敏感词的简称变形体，提出了一种基于 KMP 的汉字拆分识别算法（WS-KMP），通过分析汉字的结构对拆分后的敏感词进行模式匹配实现敏感词拆分变形体的识别。

3　敏感词变形体整理及处理方法

3.1　词的拼音模式

目前，存在很多利用具有与敏感词中读音相似的汉字来替换敏感词中汉字的情况，例如“去死（qusi）”为敏感词，恶意用户为了避开网络平台的审查，多数情况下会使用“去屎（qushi）”一词来替代。这样，不仅表达了其含义，也不会被平台所追究。为了识别出敏感词拼音的变形体，本文将敏感词与疑似敏感词转换成音码并通过编辑距离计算其相似度，判断该词是否为敏感词。

3.1.1 音码编码（soundcode SC）

音码即为汉字拼音的编码方式，能够用编码形象的表示出汉字的拼音特征，从而表示出汉字的读音特征。基于音码可将汉字的拼音转换成相应的字符序列，其结构如图 2 所示。

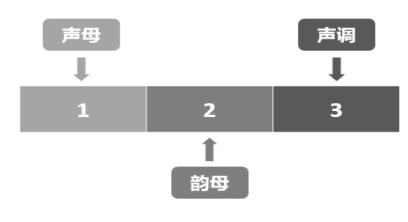


图 2　音码结构图

本文采用三位音码的的形式，利用字母对其中每一位音码进行编码。三位音码中，设第一位为声母位，第二位为韵母位，第三位为声调位。根据 1958 年第一届全国人民代表大会第五次会议批准公布推行的《汉语拼音方案》[14]，其中包含声母 23 个，韵母 34 个，为了方便编码，声调分为阴平、阳平、上声、去声和轻声。声母、韵母和声调的部分编码如表 1、表 2、表 3 所示。

表 1　声母编码表

声母	b	p	m	f	d	t	…
编码	A	B	C	D	E	F	…

表 2　韵母编码表

韵母	a	o	e	ai	ei	ao	…
编码	A	B	C	D	E	F	…

表 3　声调编码表

声调	阴平	阳平	上声	去声	轻声
编码	A	B	C	D	E

通过以上编码，可将汉字转换成一系列的字符序列，以便进行下一步的计算和比较。以“海洛因”为例，基于以上编码表得到的音码为“KDCHXDPTA”。

3.1.2 易混拼音分组

文献[15]提出了一种基于语音的字符串匹配算法，该算法用于解决英文发音相似的字符串之间的匹配问题。基于该算法，本文提出将中文易混拼音进行分组，并对所有分组赋予了一个相似性因子（该值区间为 $[0,1]$，其值越低说明字符串的相似度就越高）。

易混拼音主要分为三种：平舌音与翘舌音、边音与鼻音、前鼻音与后鼻音。表 5 是易混拼音分组的部分数据，该表的拼音分组以及相似性因子参考了文献[12]。每组相似性因子代表同组拼音被替换成同一组中的另一个拼音需要付出的“代价”，若两个拼音相同，则它们的“代价”就是 0，若两个拼音不相同，并且也没有在同一个组里面，那么他们的“代价”是 1。通过对汉语拼音的研究，部分分组情况如表 4 所示。

表 4　易混拼音分组

易混拼音	z,zh	n,l	en,eng	…
音码	S,R	G,H	J,K	…
相似性因子	0.5	0.5	0.5	…

3.1.3 基于易混拼音分组的敏感词识别算法

设有敏感词 S 和疑似敏感词 A，通过建立 unicode 的编码与汉语拼音对应的集合把汉字转换成带有声调的拼音（Pinyin4j 为汉字转拼音的 java 工具类），然后根据音码表将其转换成音码 s 和 a 且 $\text{length}(s)=i(i>0)$，

$\text{length}(a)=j(j>0)$，其编辑距为

$$edit(i,j)=\begin{cases} i, j=0 \\ j, i=0 \\ \min\begin{cases} edit(i,j-1)+1,(i>0,j>0) \\ edit(i-1,j)+1,(i>0,j>0) \\ edit(i-1,j-1)+r(s_i,t_j)\ ,(i>0,j>0) \end{cases} \end{cases}$$

采用易混拼音分组之前，函数 $r(s_i,a_j)$ 定义如下：

$$r(s_i,a_j)=\begin{cases} 0 & s_i=a_j \\ 1 & s_i\neq a_j \end{cases}$$

采用易混拼音分组之后，函数 $r(s_i,a_j)$ 定义如下：

$$r(s_i,a_j)=\begin{cases} 0 & s_i=a_j \\ 1 & s_i,a_j\text{同组} \\ 1 & s_i\neq a_j\text{且不同组} \end{cases}$$

其中：a 是字符 s_i 和 a_j 所在组的相似性因子。相似度计算公式如下：

$$\text{Similar}(s,a)=1-\frac{\text{edit}(\text{length}(s),\text{length}(a))}{\max(\text{length}(s),\text{length}(a))}$$

为了识别出敏感词的拼音变形体，本文提出基于易混拼音分组的敏感词的识别算法（以下简称为 SPGR（Similar Pinyin Grouping Recognition）算法）。算法具体的执行过程如算法 1 所示。

算法 1. SPGR algorithm

输入:敏感词 S，疑似敏感词变形体 A

输出:A 是否为 S 的拼音变形体

1.若 S.equals(A)为 true，则 S 与 A 为同一字符串，end.

2.若 S.equal(A)为 false，则根据编码表获取 S 与 A 的音码 s 与 a，且 $i=\text{length}(s)$，$j=\text{length}(a)$；

3.通过方法 $edit(i,j)$ 得到音码的编辑距离；

4.通过方法 $max(i,j)$ 获取音码长度较大的部分；

5.计算相似度 $Similar(s,a)$；

6.若 $Similar(s,a)>\theta$（θ 为阈值），则字符串 A 为字符串 S 的拼音变形体，end.

7.若 $Similar(s,a)\le\theta$，则字符串 A 不为字符串 S 的拼音变形体，算法结束，end.

以“海洛因”和“海诺因”为例，设 θ 为 90%，其拼音分别为“hailuoyin”和“hainuoyin”，转换成音码之后为“KDCHXDPTA”和“KDCGXDPTA”，采用易混拼音分组之前相似度为 88.89%，之后为 94.44%，相似度明显提高且大于 90%，则“海诺因”为“海洛因”的拼音变形体。

3.2　词的简称模式

3.2.1 词的简称

词的简称模式包括首字母缩写和词的缩写。其中首字母缩写如“法轮功”缩写为“flg”。

词的缩写的一般分为三种形式:压缩、节略和统括[16]，其中又以压缩和节略的组合生成模式最为常见。压缩，是指把全称分割为几个词语，然后从每个词语中抽取最能代表原义的汉字保留，例如“贩卖\毒品”的简称为“贩毒”;节略是指在全称中

直接省去部分词语，留下另一部分词语作为简称，例如“复旦\大学”的简称为“复旦”。压缩和节略的基本思想都是从全称中选取部分汉字或者词语重组形成简称。在重组的过程中，字序一般不会发生改变。简称中的汉字全部包含于词的全称中，因此，找到词全称的子集就可以找到其简称。如：中文字符串 $S = s_1 s_2 s_3 \cdots s_n (n > 1)$ 是由 n 个汉字构成，存在另外一个由 m 个汉字构成的字符串 $A = a_1 a_2 \cdots a_m (1 \leq m < n)$，对于任何一个字 $a_i \in A (1 \leq i \leq m)$，都有 $a_i = s_j (1 \leq i \leq m, i \leq j \leq n)$，且随着 i 值的递增，j 也呈递增趋势，则称 A 为 S 的简称。

3.2.2 敏感词的简称识别算法

为了准确识别出敏感词的简称，本文提出了敏感词的简称识别算法（以下简称为 SNR（Short Name Recognition）算法）。对于敏感词首字母缩写的识别，首先利用 Pinyin4j 获取敏感词与疑似敏感词的的首字母，然后再对首字母进行字符串的比对；对于敏感词缩写的识别，将疑似敏感词的每个字符在敏感词中按照顺序进行匹配。算法具体的执行过程如算法 2 所示。

算法 2. SNR algorithm

输入：敏感词 S，疑似敏感词变形体 A

输出: A 是否为 S 的简称变形体

1.获取字符串的首字母 S’,若 S’=A,则字符串 A 为字符串 S 的简称，end.

2.根据字符串 S 和待判断字符串 A 的有序集合 $S = (s_1 s_2 \cdots s_m)$ 和 $A = (a_1 a_2 \cdots a_n)$ 得到 $m = length(S)$，$n = length(A)$；

3.若 $m > n$,设 i,j 分别为 S,A 的下标，初始值都为 0；

4.若 $i < m \&\& j < n \&\& S_i = A_j$,则 $i{+}{+}$，$j{+}{+}$，继续执行步骤 4；

5.若 $i < m \&\& j < n \&\& S_i \neq A_j$，则 $i{+}{+}$，跳到步骤 4；

6.若 $i \geq m \,||\, j \geq n$，则跳出循环；

7.若 $j = n$,则字符串 A 为字符串 S 的简称，end.

8.若 $j \neq n$，则字符串 A 不为字符串 S 的简称，end.

以字符串“上海交通大学”，待判断字符串“交大”为例，可以得到 S=（上海交通大学），A=（交大），开始 $S_1 \neq A_1$,S 下标向右移动一个单位，继续比对直到 $S_3 = A_1$，此时 S 和 A 下标同时向右移动一个单位，$S_4 \neq A_2$,S 下标向右移动一个单位，继续比对直到 $S_5 = A_2$，此时 A 的下标等于 $length(A)$，算法结束，则“交大”为“上海交通大学”的简称。

3.3 词的拆分模式

3.3.1 汉字拆分

根据汉字的构成单位可把汉字分为独体字、合体字两类。独体字（日、月等）由笔画构成，合体字（休、取等）则由偏旁构成。现代汉字的拆分，要充分认识汉字的组成规律，要根据符合中国人书写习惯的规则来拆分。汉字的空间上的关系有：相交、相离、相接[17]。汉字的方位上的关系有：上下、左右，内外、框架、独体。

为了使每个汉字有一个全国统一的代码，我国颁布了汉字编码的国家标准：《信息交换用汉字编码字符集》[18]。区位码是一个四位的十进制数，每个区位码都对应着一个唯一的汉字或符号[19]。根据以上汉字特征对敏感词列表中的汉字进行人工拆分并采用区位码进行编码形成汉字拆分表。如表 5 所示。

表 5 汉字拆分表

汉字	区位码	拆分	区位码
法	2308	氵 去	6763 4005
秃	4526	禾几	2644 2824
国	2590	口玉	3158 5181

3.3.2 基于 KMP 的敏感词拆分识别算法

为了识别出敏感词拆分的变形体，本文提出了基于 KMP 的汉字拆分识别算法（以下简称为 WS-KMP（Word Split KMP）算法），首先根据汉字拆分表把敏感词 S 与疑似敏感词变形体 A 进行拆分并转换成相应的区位码，然后采用模式匹配 KMP 算法进行匹配。算法具体的执行过程如算法 3 所示。

算法 3. WS-KMP algorithm

输入:敏感词 S，疑似敏感词变形体 A

输出:A 是否为包含于 S 的拆分变形体

1.按照汉字拆分表将字符串 S 和待判断字符串 A 进行拆分，得到其区位码 $S = (s_1 s_2 \cdots s_m)$ 和 $A = (a_1 a_2 \cdots a_n)$；

2. $m = length(S)$，$n = length(A)$；

3.若 $m > n$，匹配不成功，end.

4.若 $m \leq n$，则 $S = (s_1 s_2 \cdots s_m)$ 为目标串，$A = (a_1 a_2 \cdots a_n)$ 为模式串，采用 KMP 算法进行匹配；

5.若存在 $s_1 s_2 \cdots s_m = a_i a_{i+1} \ldots a_{m+i-1}$,匹配成功，end.

6.若不存在 $s_1 s_2 \cdots s_m = a_i a_{i+1} \ldots a_{m+i-1}$，匹配失败，end.

以敏感词“海洛因吗啡”和疑似敏感词“口马口非”为例，首先将两个词进行拆分，结果为“氵 每氵 各口大口马口非”和“口马口非”，获取相应的区位码 S=（6763353167632487315820833158347731582339），A=（3158347731582339），采用模式匹配 KMP 算法进行匹配。首先 $S_1 \neq A_1$，S 下标每次向右移动一位继续比对直到 $S_4 = A_1$，然后 S 与 A 下标同时向右移动一位 $S_5 \neq A_2$，根据 $Next(j)$ 函数确定每次 A 的下标，直到 $s_{25} s_{26} \cdots s_{40} = a_1 a_2 \cdots a_{16}$，则匹配成功，所以 A 是包含于 S 的拆分变形体。

4 实验与分析

本实验在具有 2.4GHz Inter(R)Core^{TM} i5 处理器 8GB 内存的机器上运行，操作系统为 windows 10，编程语言为 Java。

4.1 数据集

为了评估面向中文敏感词变形体的识别方法的效果,从搜狗实验室(http://www.sogou.com/labs/)的 SogouCA(版本：2012)全网新闻数据库中随机抽取了含有疑似敏感词的 800 篇新闻文本(包含科技、体育、金融、社会、娱乐等题材)作为测试数据集。对数据集中的敏感词及其变形体进行人工的识别和分类，共发现 67 个敏感词及其变形体 400 个，涵盖了词的拼音、词的简称、词的拆分三种变形体情况，并将识别出的敏感词存入敏感词表中。在实验中，首先对以上 67 个敏感词进行人工拆分，再

将敏感词变形体的数据集随机分成 5 组进行测试，第一组为 80 个，第二组为 160 个，第三组为 240 个，第四组为 320 个，第五组为 400 个。数据集中所抽取的敏感词变形体的部分举例如表 6 所示。

表 6 敏感词及其变形体举例

敏感词	变形体		
	词的拼音	词的简称	字的拆分
法轮功	falungong	flg	氵去车仑工力
兴奋剂	xingfenji	xfj	兴大田齐 刂
贩卖毒品	fanmaidupin	贩毒 fmdp	贝反卖毒口口口
袭警	xijing	xj	龙衣敬言

4.2 实验分析

在实验中，通过算法的准确率、查全率和运行时间三个方面来验证敏感词变形体识别算法的有效性。设词的拼音模式中 θ 为 91%，将识别结果分为四种情况：识别为真，实际为真（TP）；识别为假，实际为假（TN）；识别为真，实际为假（FP）；识别为假，实际为真（FN）。其中，查全率=TP/（TP+FP），准确率=TP/（TP+FN）。实验结果如表 7 所示。

表 7 实验结果

数据集大小	基于变形体的中文字符串识别算法					
	TP	FP	FN	查全率	准确率	运行时间（s）
80	71	6	9	88.75%	92.22%	0.95
160	135	11	25	84.38%	92.47%	1.11
240	211	19	29	87.91%	91.73%	1.87
320	285	21	35	89.06%	93.13%	2.53
400	357	24	43	89.25%	93.70%	3.46

通过上述实验结果可以计算出基于变形体的中文字符串识别算法的平均查全率和准确率分别为 87.87%、92.65%，在查全率方面，五组数据集中最低查全率为 84.38%，导致该组结果偏低的原因可能是由于被识别的敏感词及其变形体实际情况比较复杂，本文所提出的识别规则还需进一步的改进和优化；随着数据量的增加查全率基本稳定在 89%；在准确率方面五组数据集均高于 91%，且最高准确率接近 94%；在运行时间方面，随着数据量的增加，运行时间也有相应的增加。综上所述，本文所提出算法在实验中均已得到了较为理想的效果。

4.3 实验结论

通过对不同的数据集进行实验，根据上述实验结果可以发现：本文提出的算法在查全率、准确率和运行时间的指标上均有较稳定的体现，证明了所提出算法的可行性和有效性。

5 结束语

本文所提出的方法能够根据词的拼音、词的简称和词的拆分三种中文字符的变形体识别出其相关的敏感词，辅助提高了审查的准确率和效率。实验证明本文所提出的方法可较为准确的识别出中文字符串的各种变形形式，有效提高敏感词审查的准确度，其效果也更接近于人脑识别的结果。但是词的拆分需要人工进行操作，当敏感词较多的时候，工作量会比较大，所以汉字的自动拆分是非常重要的，这也是下一步的工作。

参考文献：

[1] 李扬，潘泉，杨涛. 基于短文本情感分析的敏感信息识别 [J]. 西安交通大学学报, 2016, 50 (9): 80-84.

[2] 叶蕾，郘国奇，肖健. 模糊遗传算法在敏感词分类优化中的应用 [J]. 计算机应用研究, 2012, 29 (7): 2549-255.

[3] 薛朋强，努尔布力，吾守尔·斯拉木. 基于网络文本信息的敏感信息过滤算法 [J]. 计算机工程与设计, 2016, 37 (9): 2447-2452.

[4] Yoon Taijin, Park Sunyoung, Cho H G. A smart filtering system for newly coined profanities by using approximate string alignment [C]// Proc of IEEE International Conference on Computer & InformationTechnology. 2010: 643-650.

[5] Sood S 0, Antin J, Churchill E F. Using crowdsourcing to improve profanity detection [J]// Proc of AAAI Spring Symposium Series. 2012: 69-74.

[6] Ghauth K I, Sukhur M S. Text censoring system for filtering malicious content using approximate string matching and Bayesian filtering [C]// Computational Intelligence in Information Systems. Springer International Publishing, 2015: 331: 149-158

[7] 李少卿，吴承荣，曾剑平，等. 不良文本变体关键词识别的词汇串相似度计算 [J]. 计算机应用与软件, 2015 (3): 151-157.

[8] Levenshtein V I. Binary codes capable of correcting deletions, insertions and reversals [J]. Soviet Physics Doklady, 1966, 10 (1): 707-710.

[9] Liu Baoyan, Lin Hongfei, Zhao Jing. Chinese sentence similarity computing based on improved edit-distance and dependency grammar [J]. Computer Applications & Software, 2008.

[10] 李圣文，凌微，龚君芳，等. 一种基于熵的文本相似性计算方法 [J]. 计算机应用研究, 2016, 33 (3): 665-668.

[11] Knuth D, Morris J, Pratt V. Fast pattern matching in strings [J]. SIAM Journal on Computing, 1977, 6 (2): 323-350.

[12] Chang Jingshin, Teng Weilun. Mining atomic Chinese abbreviation pairs: a probabilistic model for single character word recovery [J]. Language Resources and Evaluation, 2007, 40 (3//4): 367–374

[13] 李钝，曹元大，万月亮. 信息安全中的变形关键词的识别 [J]. 计算机工程, 2007, 33 (21): 55-156.

[14] 中国文字改革委员会. 汉语拼音方案 [S]. 中国：中国文字改革委员会, 1967

[15] Zobel J, Dart P. Phonetic string matching: Lessons from information retrieval [C]// Proc of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. 1996: 166-172

[16] 殷志平. 构造缩略语的方法和原则 [J]. 语言教学与研究, 1999 (2): 73-82

[17] 朱文轩. Blog 文本内容敏感信息的自动提取技术 [D]. 上海: 上海交通大学, 2008

[18] 中国国家标准总局. 信息交换用汉字编码字符集 [S]. 中国: 中国国家标准总局, 1980

[19] 杨超, 谢剑刚. 基于区位码字典对数控程序进行中文注释 [J]. 中国科技信息, 2015 (17): 65-66